

## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

**NASA TECHNICAL  
MEMORANDUM**

**NASA TM X-72749**

**NASA TM X-72749**

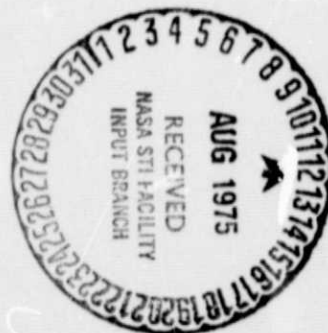
(NASA-TM-X-72749, A PROGRAM FOR CONTOURING  
RANDOMLY SPACED DATA (NASA) 35 p HC \$3.75  
CSCL 09E

N75-28748

Unclas  
G3/61 31101

A PROGRAM FOR CONTOURING RANDOMLY SPACED DATA

By Roy W. Hamm, James F. Kibler, and W. Douglas Morris



This informal documentation medium is used to provide accelerated or special release of technical information to selected users. The contents may not meet NASA formal editing and publication standards, may be revised, or may be incorporated in another publication.

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
LANGLEY RESEARCH CENTER, HAMPTON, VIRGINIA 23665**

1. Report No. NASA TM X-72749		2. Government Accession No.		3. Participant's Catalog No.	
4. Title and Subtitle A PROGRAM FOR CONTOURING RANDOMLY SPACED DATA				5. Report Date JULY 1975	
				6. Performing Organization Code	
7. Author(s) * Roy W. Hamm, James F. Kitler **, and W. Douglas Morris **				8. Performing Organization Report No.	
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665				10. Work Unit No.	
				11. Contract or Grant No.	
				13. Type of Report and Period Covered Technical Memorandum	
12. Sponsoring Agency Name and Address National Aeronautics & Space Administration Washington, DC 20546				14. Sponsoring Agency Code 161-03-03-01	
15. Supplementary Notes * Computer Sciences Corporation ** NASA-Langley Research Center					
16. Abstract  This paper presents a brief description of a digital computer program which prepares contour plots of three dimensional data. The contouring technique uses a triangulation procedure developed by Dr. C. L. Lawson of the Jet Propulsion Laboratory. As presently configured, the program can accept up to 56,000 randomly spaced data points, although the required computer resources may be prohibitive. However, with relatively minor internal modifications, the program can handle essentially unlimited amounts of data. Up to 20 contouring intervals can be selected and contoured with either polygonal lines or smooth curves. Sample cases are illustrated. A general description of the main program and primary level subroutines is included to permit simple modifications of the program.					
17. Key Words (Suggested by Author(s)) (STAR category underlined) Contour Randomly spaced data <u>Computer Programming and Software</u>				18. Distribution Statement Unclassified Unlimited	
19. Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 35	22. Price* \$3.75

\*Available from { The National Technical Information Service, Springfield, Virginia 22151  
STI/NASA Scientific and Technical Information Facility, P.O. Box 33, College Park, MD 20740

# A PROGRAM FOR CONTOURING RANDOMLY SPACED DATA

By Roy W. Hamm, James F. Kibler, and W. Douglas Morris

Langley Research Center

## SUMMARY

This paper presents a brief description of a digital computer program which prepares contour plots of three-dimensional data. As presently configured, the program can accept up to 56,000 randomly spaced data points, though the required computer resources may be prohibitive. However, with relatively minor internal modifications, the program can handle essentially unlimited amounts of data. The contouring techniques use a triangulation procedure developed by Dr. C. L. Lawson of the Jet Propulsion Laboratory. The contour points can be fitted with a smooth curve using an interpolating spline under tension. Sample cases illustrate contours of ocean wave simulation data. A general description of the main program and primary level subroutines is included to permit simple modifications of the program to handle other applications.

## INTRODUCTION

Large amounts of three-dimensional data have been generated as a result of parametric studies of wave conditions off the Mid-Atlantic Coast (Ref. 1). One way to analyze the data is by contouring wave height and wave orbital velocity parameters at selected values. In this way, the areas of differing wave activities can be defined. This information, in turn, is valuable in locating future offshore facilities in areas of minimum wave activity, or for

locating dumping sites for either maximum or minimum dispersion of the dumped effluent or spoil. However, because of the large number of data points (80,000 generated by the study) and since these points do not fit into a rectangular grid pattern, existing Langley Research Center (LRC) contouring programs cannot adequately handle the data.

In order to satisfy these requirements, a digital computer program has been modified to contour wave data. The program was originally developed by Dr. C. L. Lawson of the Jet Propulsion Laboratory. His "Triangulation" technique allowed the contouring of randomly spaced input data, without first fitting the data into a rectangular grid (required by most other programs). Later modifications were made by Dr. Alan K. Cline of ICASE\*. One of the modifications incorporated a spline under tension to fit a smooth curve to the contour points. This program was then adapted to the Langley Research Center CDC 6600 system. Further modifications have been made which reduce the memory storage, execution time, and increase the plotting efficiency. The modified contouring program represents an improvement over models presently available at LRC because it can provide smooth contouring of a relatively large number of randomly spaced data points (56,000). With minor internal modifications, the program could handle larger amounts of data.

This paper presents a brief description of the contouring technique, a description of the computer program, and sample results from four typical cases.

\*Institute for Computer Applications in Science and Engineering, NASA-Langley Research Center

## Description of Contouring Technique

A contour plot is a convenient way to display a function of two independent variables. The contour is simply a line connecting constant values of the function plotted on a grid composed of the corresponding independent variables. In this way, a three-dimensional surface can be easily visualized in two dimensions. This section briefly describes the triangulation contouring technique.

Assume that the function to be contoured is available as an array of randomly spaced data points. Each data point consists of an  $X, Y$  and  $Z$  value. The  $X$  and  $Y$  values correspond to the independent coordinates to be plotted and  $Z = Z(X, Y)$  is the function value to be contoured. The contouring process begins by forming a convex region composed of triangles from the set of  $X-Y$  data points (fig. 1). Figure 2 illustrates an arrangement of triangles which form a convex region from the data points in figure 1. Obviously, this arrangement is not unique. In order to resolve this problem, a criterion is established to maximize the smallest interior angle between possible pairs of triangles. In addition, the criterion is

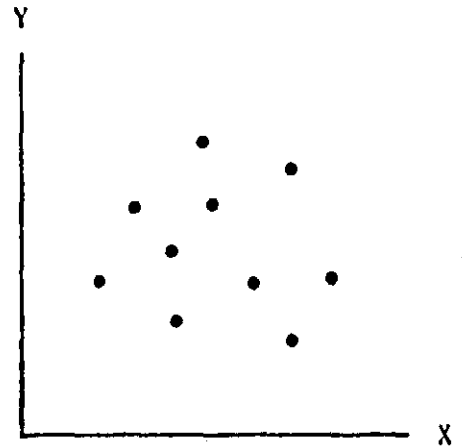


Figure 1.- A set of input data points.

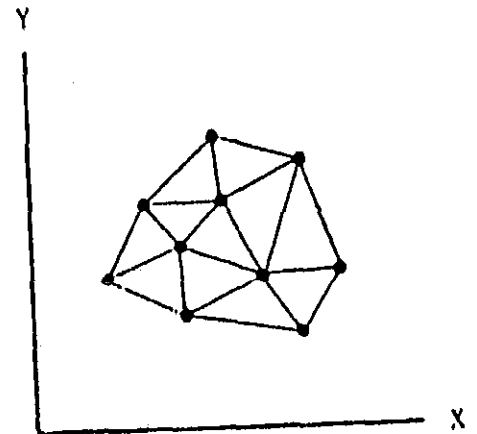


Figure 2.- A convex region composed of triangles.

required to more evenly distribute interpolated contour points. For example, given a set of four data points, there are two choices for arranging the two triangles (figure 3). Under the above criterion, 3-b is chosen as the preferred arrangement.

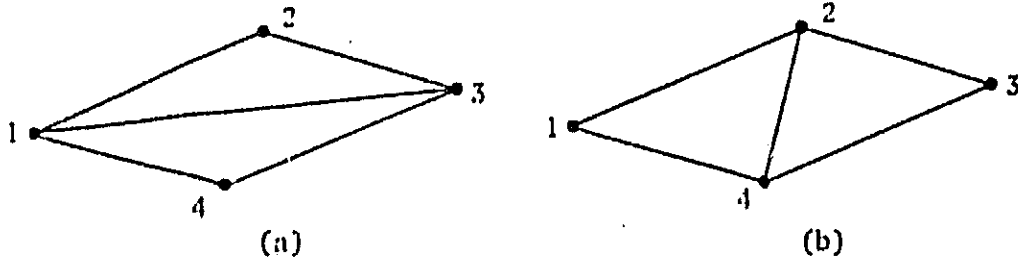


Figure 3.- Two choices for arranging triangles for four data points.

After the triangles are arranged, each  $Z$  value is assigned to its corresponding  $X$ - $Y$  vertex in each triangle. An interpolating plane which fits the vertices exactly is then calculated for each triangle. If a chosen contour level lies between the  $Z$  values for any triangle, the two contour points which lie on the sides of the triangle are computed from the interpolating plane (fig. 4). These contour points are calculated for each triangle in the convex region and for each specified contour level. The contour points for any given contour level may then be connected by straight lines to yield a polygonal approximation to the required contour. Alternatively,

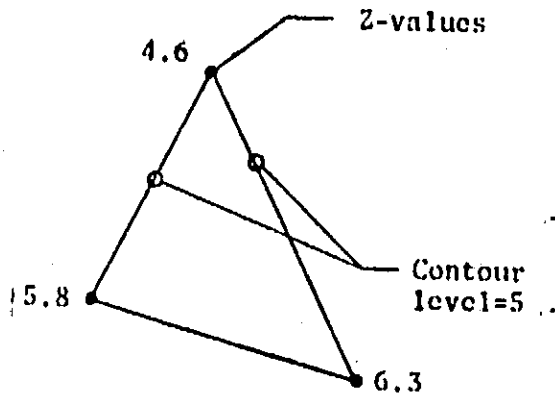


Figure 4.- A triangle with linearly interpolated contour points.

be used to fit a smooth curve to each set of contour points. This curve fits the interpolated contour points exactly and varies smoothly between them. The tension factor may be adjusted to yield suitable curves (a large factor corresponds to straight lines between contour points). The contours which results from this technique are, of course, only approximate. Their accuracy is limited by the linear interpolating plane procedure and is dependent upon the density of the available data.

## PROGRAM DESCRIPTION

### General

A digital computer program has been developed to implement the contouring technique. Program CONTOUR is coded in FORTRAN for a CDC 6600 computer using the LRC operating system. Plotting is done using CALCOMP routines and any of the graphics post-processors available at LRC. The main program handles input and controls primary-level subroutines which contour the data. A brief description of the primary-level subroutines is presented in the appendix. A listing and flow chart of the main program are given in figures 5 and 6. Data transfer is through labeled COMMON and subroutine argument lists. Table I describes the COMMON blocks.

### Program Input

The program is controlled by three things:

1. Tape input is used for the data points to be contoured.
2. Card input defines the program logic.
3. Field length controls the maximum problem size.

The data points to be contoured are read from a magnetic tape or disk file identified as TAPE 9. The data tape must have been created using the LRC output routine RECOUT. The user should prepare TAPE 9 with one record for each data point. Each record contains the following:

- X - X-coordinate of system
- Y - Y-coordinate of system
- Z - Value to be contoured
- V - Alternate value to be contoured for same X and Y (if there is none, set  $V = 0$ ).

Card input to the program consists of a problem title card and NAMELIST input. The problem title card is read with a 3 A10 format and is required only once for each run. The NAMELIST input is identified as CASE and each successive contour plot requires a separate NAMELIST input. Default values and descriptions of each variable of the NAMELIST are contained in Table II. If the user desires to use default values, only two variables must be specified in the NAMELIST. They are NCL which is the number of contour levels, and CL which is a vector containing the actual values of each contour level. NCL is presently limited to a maximum of 20. The remaining variables in the NAMELIST do not have to be specified unless the user desires to change the default values.

In order to reduce core storage requirements, the program partitions the data to be contoured. This partitioning is handled internally and is controlled by three variables--IMAX, MAXPTS, and LPCT. IMAX is the maximum number of points in each partition. MAXPTS is the total number of points to be plotted (equal to the number of records on TAPE 9 if no records are skipped. Note--

MAXPTS = 0 yields a plot of the complete file). LPCT is the percentage of overlap between partitions in order to yield continuity between partitions. Using default values, the program will place a maximum of 1000 data points in each partition with a 20 percent overlap, and the complete file will be plotted.

The field length required for CONTOUR is dependent upon the number of data points to be contoured. The program requires 42K octal storage locations. To this must be added the larger of 24 times IMAX, or four times MAXPTS plus 2K octal. Thus, the user may trade-off IMAX and MAXPTS to reduce core storage.

Several options are available to the user. IZPLOT controls the choice of which value in the data record is to be contoured (Z or V). IPLOT controls whether contours are drawn by polygonal lines, or by a spline under tension (smooth curves). And NDEC and HTNUM control the contour level labels which may be suppressed if desired.

To terminate the program, an end of file (EOF) is checked on card input. If an EOF is detected, the plot file is closed and the program is terminated.

#### Program Output

Program output consists of user messages and the contour plots. The plots are generated by one of the LRC graphics post-processor (e.g., VARIAN). Each NAMELIST is printed, along with the total number of points plotted and the partition sizes. Data error messages are provided. The error messages are contained in the primary-level subroutine descriptions in the appendix.

### Sample Results

Four sample plots were generated and are presented in figures 7 through 10. The data used for the plots are typical of computed results from a parametric study of wave conditions. In general, default values of the NAMELIST were used and the NAMELIST required for each plot is given in the figures.

The figures are identified as follows:

Figure 7: One contour (NCL = 1) of wave height (IZPLOT = 1) smoothed with spline under tension (IPLOT = 1).

Figure 8: One contour of orbital velocity (IZPLOT = 2) smoothed with spline under tension.

Figure 9: Three contours (NCL = 3) of wave height smoothed with spline under tension.

Figure 10: One contour of wave heights with polygonal lines (IPLOT = 0).

For this sample case, there were 3500 data points. The points were split into partitions with a maximum size of 1000 points and 20 percent overlap. The case required a storage of 121000 octal locations and a run time of 500 CPU seconds. These values will vary with problem size.

### CONCLUDING REMARKS

A contouring technique has been presented which can contour randomly spaced input data. A digital computer program has been developed at the Langley Research Center which implements this technique. The program can handle up to 56,000 data points, and provide for up to 20 contour intervals for a multiple number of parameters. Contour lines can be either polygonal

or smooth, using a spline under tension. With relatively minor internal modifications, the program can handle essentially unlimited amounts of data from a wide range of applications.

#### REFERENCES

1. Goldsmith, Victor; Morris, W. Douglas; Byrne, Robert J.; and Whitlock, Charles H.: Wave Climate Model of the Mid-Atlantic Shelf and Shoreline (Virginian Sea). NASA SP-358, 1974.
2. Schweikert, D. C.: An Interpolation Curve Using a Spline in Tension. J. Math. Phys. 45, 1966, pp. 312-317.

PRECEDING PAGE BLANK NOT FILMED

TABLE I.- DESCRIPTION OF ELEMENT IN LABELED COMMON BLOCK

Labeled Common Name	Variable Name	Description
PARAM	NB	Number of boundary points of the triangular grid
	NPTS	Number of points of the triangular grid
	XSCALE	X scale factor
	YSCALE	Y scale factor
	XMIN	Minimum X value
	YMIN	Minimum Y value
	XMAX	Maximum X value
	YMAX	Maximum Y value
	NL	Number of lines of the triangular grid
	NT	Number of triangles of the triangular grid
	NI	Number of interior points of the triangular grid
SIGI	SIGMA	Tension factor applied to cubic spline
CONTRL	CLEVEL	Contour level value
	NDEC	Number of digits to the right of decimal point on the contour values
	HTN	Scaled heights of plotted contour level digits
FLAGS	IPLOT	Type of smoothing
	ISKIP	Number of records skipped on the input file

TABLE I.- CONTINUED

Labeled Common Name	Variable Name	Description
BNDPLT	IZPLOT	Plot type flag
	IMAX	Maximum points per partition
	MAXPTS	Maximum points plotted
	LPCT	Plot percentage overlap
	XBMIN	Minimum boundary of the partitioned plot
	YBMAX	Maximum boundary of the partitioned plot
	XO	Previous plotted X value
	YO	Previous plotted Y value
RAF	IPEN	Pen position of the plot
	INDEX(101)	Random Access File index
	LNG(20)	Record length for each partition
	BND(2,20)	Plot boundaries for each partition
	NBI(2,20)	Number of boundary and interior points of each partition
	IFET(18)	File Environment Table for the Random Access File
TMCOM2	NBLK	Number of partitions to be plotted
	LL(4)	Communication indices of the line array

TABLE II.- DESCRIPTION OF ELEMENTS IN NAMELIST CASE

Variable Name	Default Value	Description
NCL	-	Number of contour levels (maximum of 20)
CL	-	Actual values of contour levels (dimensioned 20)
IPLLOT	1	= 0 yields polygonal line contour = 1 yields smooth contours (spline with tension)
IS&IP	0	Number of records to skip between input data points
IZPLOT	1	= 1 plot of z-values = 2 plot of v-values
IMAX	1000	Maximum number of points to be plotted per partition
MAXPTS	0	Maximum total points to be plotted = 0 Plot the complete file
NDEC	- 1	Number of digits to the right of decimal point on the contour = - 1 Suppresses decimal
HTNUM	.0.15	Height of plotted contour level digits (inches) = 0 Suppresses contour numbers
LPCT	20	Percent overlap of the plot partitions

TABLE III - DESCRIPTION OF SECONDARY SUBROUTINES

Subroutine Name	Description
REQFL	Finds out present program field length or requests a larger or smaller field length.
INITFET	Initializes a File Environment Table.
ADDFILE	Adds a file name to the communications area.
SUBFILE	Subtracts a file name from the communications area.
LTEST	Computes the quadrant of a point.
STEST	Tests two triangles to maximize the minimum angles.
KURV1	Computes parameters necessary for a spline under tension for an open curve.
KURV2	Computes the points for a spline under tension for an open curve.
KURVP1	Computes parameters necessary for a spline under tension for a closed curve.
KURVP2	Computes the points for a spline under tension for a closed curve.
FRSTPT	Moves plotter to the first point of a line.
VECTOR	Moves plotter to successive points of a line.
POINT	Plots a single point.

TABLE IV - DESCRIPTION OF SYSTEM ROUTINES

<u>Routine Name</u>	<u>Description</u>
RECIN	Reads binary records written by the LRC routine RECOUT.
OPENMS	Initializes a random access file.
WRITMS	Writes a random access file.
READMS	Reads a random access file.
EVICT	Deletes a file from the system and releases the disk space.
LOCF	Defines the address of a word in core.

```

PROGRAM CONTOUR(INPUT=1,OUTPUT=1,TAPE5=INPUT,TAPE6=OUTPUT)
DIMENSION CODE(3),CL(20)
COMMON /XTXRXI/ NB,NPTS,XSCALE,YSCALE,XMIN,YMIN,XMAX,YMAX,NL,NT,NI
COMMON /SIGI/ SIGMA
COMMON /CONTRL/ CLEVEL,NDEC,HTN
COMMON /FLAGS/ IPLOT,ISKIP,IZPLOT,IMAX,MAXPTS,LPCT
COMMON /HNDPLT/ XRMIN,XRMAX,XO,YO,JPEN
COMMON /RAF/ INDEX(10),LNG(20),BND(2,20),NB(12,20),IFET(18),NBLK
COMMON A(1)

```

```

*      DYNAMIC STORAGE IS USED FOR STORAGE OF XYZ POINTER,LINE AND
*      TRIANGLE ARRAYS' SO PROGRAM FIELD LENGTH DEFINES CORE STORAGE

```

```

*      FL = 42K OCTAL + MAXIMUM(24*(MAX,4*MAXPTS) DECIMAL

```

```

*      NAMELIST INPUT

```

```

*      NCL      -NUMBER OF CONTOUR LEVELS (MAX = 20)
*      CL       -ACTUAL VALUES OF CONTOUR LEVELS
*      IPLOT    -=0 YIELDS POLYGONAL LINE CONTOUR
*              -=1 YIELDS SMOOTH CONTOURS (SPLINE WITH TENSION)
*      ISKIP    -NUMBER OF RECORDS TO SKIP BETWEEN ACTUAL DATA PTS
*      IZPLOT   -=1 WAVE HEIGHT PLOT
*              -=2 ORBITAL VEL PLOT
*      IMAX     -MAXIMUM NUMBER OF POINTS TO BE PLOTTED IN ONE BLK
*      MAXPTS   -MAXIMUM TOTAL POINTS TO BE PLOTTED
*              =0 PLOT COMPLETE FILE
*      NDEC     -NUMBER OF DIGITS TO RIGHT OF DECIMAL POINT ON
*              CONTOUR VALUES (-1 SUPPRESSES DECIMAL)
*      HTNUM    -HEIGHT OF PLOTTED CONTOUR LEVEL DIGITS (INCHES)
*              =0 SUPPRESSES CONTOUR NUMBERS
*      LPCT     -PERCENT OVERLAP OF THE PARTITIONS

```

```

*      RANDOM ACCESS FILE STRUCTURE

```

```

*      RECORD 1      FIRST BLOCK X
*      RECORD 2      FIRST BLOCK Y
*      RECORD 3      FIRST BLOCK Z - WAVE HEIGHT
*      RECORD 4      FIRST BLOCK V - ORBITAL VEL
*      RECORD 5      SECOND BLOCK X
*      RECORD 6      SECOND BLOCK Y
*
*      .
*      .
*      RECORD 4*NBLK-3 LAST BLOCK X
*      RECORD 4*NBLK-2 LAST BLOCK Y
*      RECORD 4*NBLK-1 LAST BLOCK Z - WAVE HEIGHT
*      RECORD 4*NBLK  LAST BLOCK V - ORBITAL VEL
*      RECORD 4*NBLK+1 FIRST BLOCK LINE AND ITR! ARRAYS
*      RECORD 4*NBLK+2 SECOND BLOCK LINE AND ITR! ARRAYS
*
*      .
*      .
*      RECORD 5*NBLK  LAST BLOCK LINE AND ITR! ARRAYS

```

```

*      PROGRAM IS STRUCTURED FOR A MAXIMUM OF TWENTY PARTITIONS
*      WHERE NBLK = (100*(MAXPTS-IMAX))/((100-LPCT)*IMAX)+1

```

```

*      NAMELIST /CASE/ NCL,CL,IPLOT,ISKIP,IZPLOT,IMAX,MAXPTS,NDEC,HTNUM
*      ,LPCT

```

```

*      INITIALIZE NAMELIST

```

```

*      IMAX = 1000
*      MAXPTS = 0
*      NDEC = -1
*      ISKIP = 0
*      IZPLOT = 1
*      IPLOT = 1
*      HTNUM = 0.15
*      LPCT = 20

```

```

*      INITIALIZE PROGRAM

```

```

*      SIGMA = -3
*      SC = 20.0
*      XD = SC*21.0
*      YD = SC*10.0
*      TB = SC
*      TS = 2.0/SC
*      HT = SC*0.3
*      XS = SC*3.0
*      YS = SC*2.0
*      DV = 10.0/SC
*      IUSED = 1000000
*      DIV = 1.0E-10
*      R = URAN(.3141591928374)
*      IFLAG = 0

```

```

*      INITIALIZE THE PLOTTER

```

ORIGINAL PAGE IS  
OF POOR QUALITY

Figure 5. Listing of main program

```

CALL PSEUDO
CALL LEROY
*
*   PROBLEM TITLE
READ (5,200) CODE
200 FORMAT(3A10)
10 CONTINUE
*
*   READ NAMELIST
READ(5,CASE)
IF (FOF(5) 20,30
20 CONTINUE
*
*   CLOSE PLOT
CALL CALPLT(0.0,999)
STOP
30 CONTINUE
WRITE(6,CASE)
HTN = SC*HTNUM
*
*   SET UP AXES
CALL CALPLT(1.0,1.0,-3)
CALL NOTATE(-40.0,0.0,HT,CODE,90.0,30)
CALL AXES(0.0,0.0,XD,0.0,DV,TB,TS,1H,-4T,-1)
IF (IZPLOT,EQ,2) GO TO 40
CALL AXES(0.0,90.0,YD,0.0,DV,TB,TS,20H WAVE HEIGHT (METERS),HT,20)
GO TO 50
40 CONTINUE
CALL AXES(0.0,90.0,YD,0.0,DV,TB,TS,20H ORBITAL VEL (CM/SEC),HT,20)
50 CONTINUE
CALL GRID(0.0,XS,YS,7,5)
IF (IFLAG.GT.0) GO TO 70
IFLAG = 1
*
*   READ THE INITIAL POINT AND WRITE ON RAF
CALL READIN
*
*   LOOP ON THE NUMBER OF BLOCKS
DO 60 I=1,NBLK
IREC = 4*(I-1)+1
N = LNG(I)
*
*   LOCATE ALL ARRAYS IN CORE
LOCKX = 1
LOCKY = LOCKX+N
LOCP = LOCKY+N
LOCL = LOCP+N+2
*
*   READ X AND Y FROM RAF
CALL READMS(9,A(LOCKX),N,IREC)
IREC = IREC+1
CALL READMS(9,A(LOCKY),N,IREC)
*
*   COMPUTE THE TRIANGULAR GRID
CALL TRIANG(A(LOCKX),A(LOCKY),N,A(LOCP),ISUP,IUSED)
*
*   WRITE THE LINE AND TRIANGLE DATA TO RAF
IREC = 4*NBLK+1
NB1(1,1) = NB
NB1(2,1) = NI
NLC = 2*NB+3*(N-NB-1)
NTC = NB+2*(N-NB-1)
LENGTH = 5*(NLC+1)+3*(NTC+1)
CALL WRITMS(9,A(LOCL),LENGTH,IREC)
60 CONTINUE
70 CONTINUE
*
*   LOOP ON THE BLOCKS
DO 100 I=1,NBLK
IREC = 4*(I-1)+1
*
*   THE LENGTH FOR THE XYZ RECORDS
N = LNG(I)
*
*   LOCATE THE ARRAYS IN CORE
LOCKX = 1
LOCKY = LOCKX+N
LOCKZ = LOCKY+N
LOCL = LOCKZ+N
*
*   READ THE X ARRAY
CALL READMS(9,A(LOCKX),N,IREC)
IREC = IREC+1
*
*   READ THE Y ARRAY
CALL READMS(9,A(LOCKY),N,IREC)
IREC = IREC+IZPLOT
*

```

```

*      READ THE Z OR V ARRAY
CALL READMS(9,A(LOCZ),N,IREC)

*
*      PLOTTING BOUNDARIES FOR THE PARTITION
XBMIN = BND(1,1)
XBMAX = BND(2,1)

*
*      DEFINE THE NUMBER OF BOUNDARY POINTS NB
*      INTERIOR POINTS NI
*      GRID LINES NL
*      GRID TRIANGLES NT

NB = NB1(1,1)
NI = NB1(2,1)
NL = 2*NB+3*(NI-1)
NT = NB+2*(NI-1)
NLC = 2*NB+3*(NI-NB-1)
NTC = NB+2*(NI-NB-1)
LENGTH = 5*(NLC+1)+3*(NTC+1)

*
*      READ THE LINE AND TRIANGLE ARRAYS
IREC = 4*NBLK+1
CALL READMS(9,A(LOCL),LENGTH,IREC)
IF (1PLOT.GT.0) GO TO 90

*
*      POLYGONAL LINE CONTOUR PLOTS
ZSPEC = 1.0E+9
CALL TRIORD(A(LOCZ),N,A(LOCL),ZSPEC)
DO 80 J=1,NCL
CLEVEL = CL(J)
CALL SCAN(A(LOCX),A(LOCY),A(LOCZ),N,A(LOCL),CL(J))
80 CONTINUE
GO TO 100
90 CONTINUE

*
*      SMOOTH CONTOUR PLOTS (SPLINE WITH TENSION)
LOCT = LOCL+5*(NLC+1)
CALL CONTR(A(LOCX),A(LOCY),A(LOCZ),A(LOCL),A(LOCT),CL,NCL,MODE)
IF (MODE.EQ.1) GO TO 100
WRITE(6,201) MODE
201 FORMAT(* ---CONTR1 ERROR. MODE ==12)
100 CONTINUE

*
*      NEXT FRAME
CALL NFRAME(28,0,0)
GO TO 10
END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

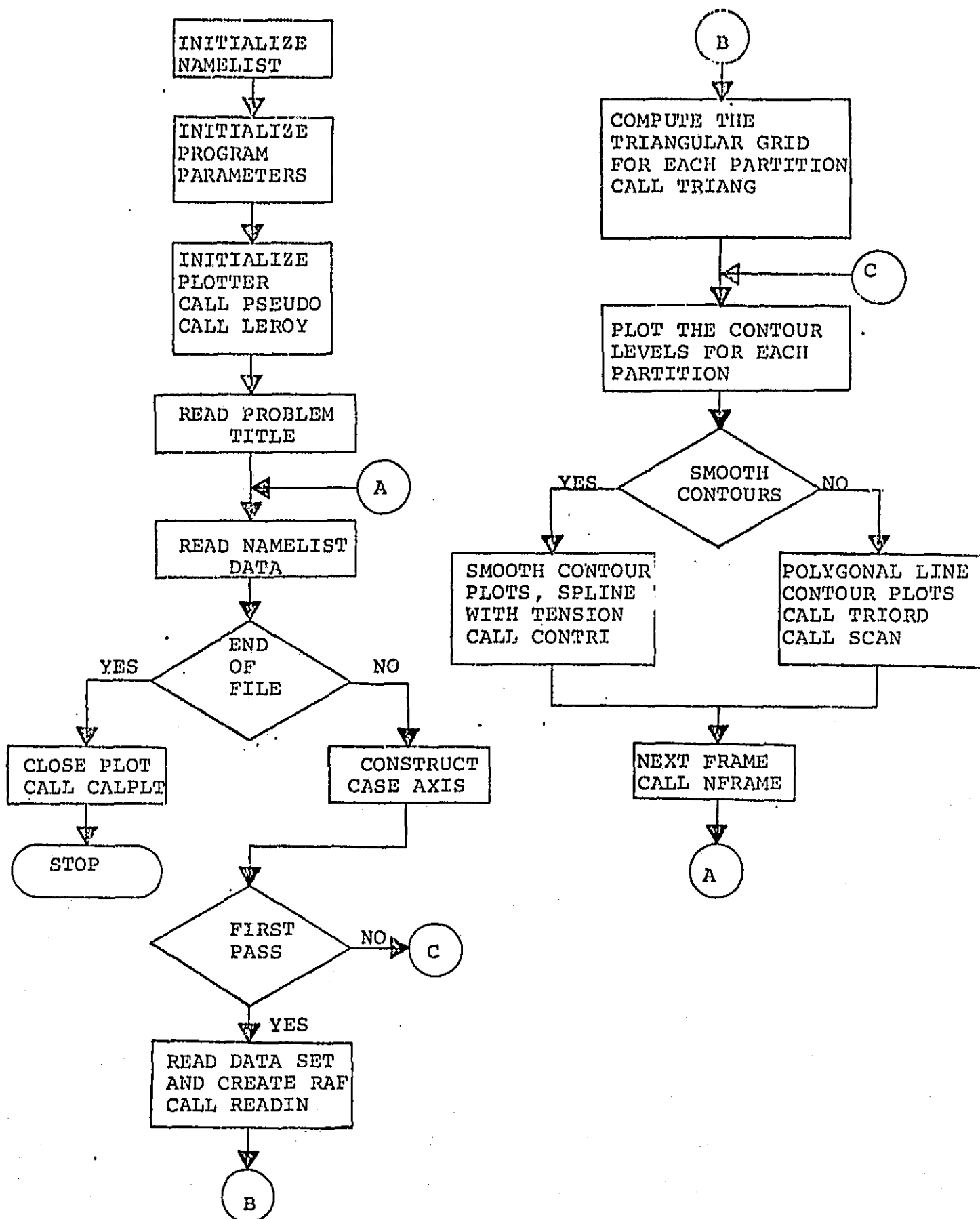


Figure 6. Flow chart of main program

Namelist input:

```

$CASE
NCL=1,
CL=0.1E+01,
IPLOT=1,
ISKIP=0,
IZPLOT=1,
IMAX=1000,
MAXPTS=0,
NDEC=-1,
HTNUM=0.15E+00,
$END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

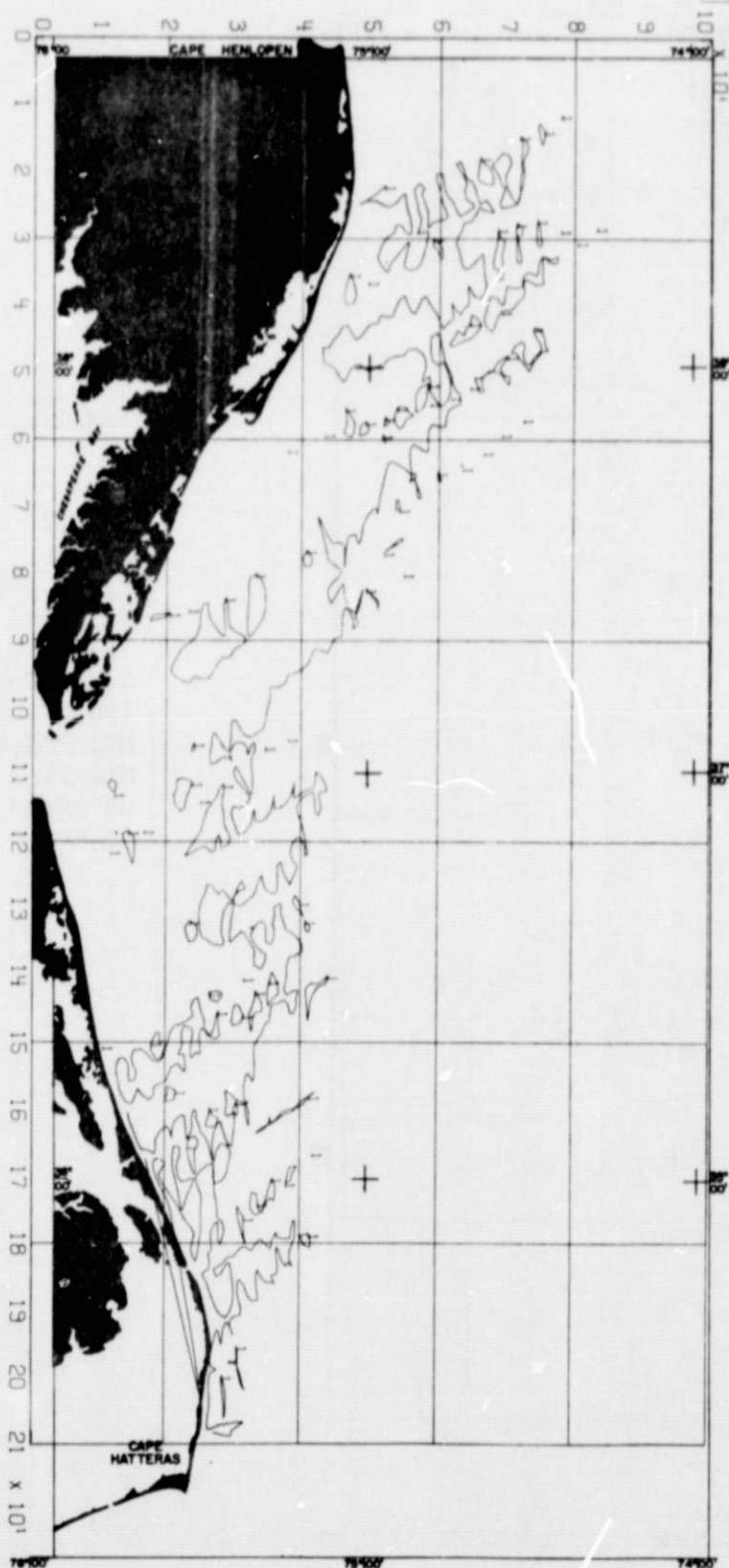
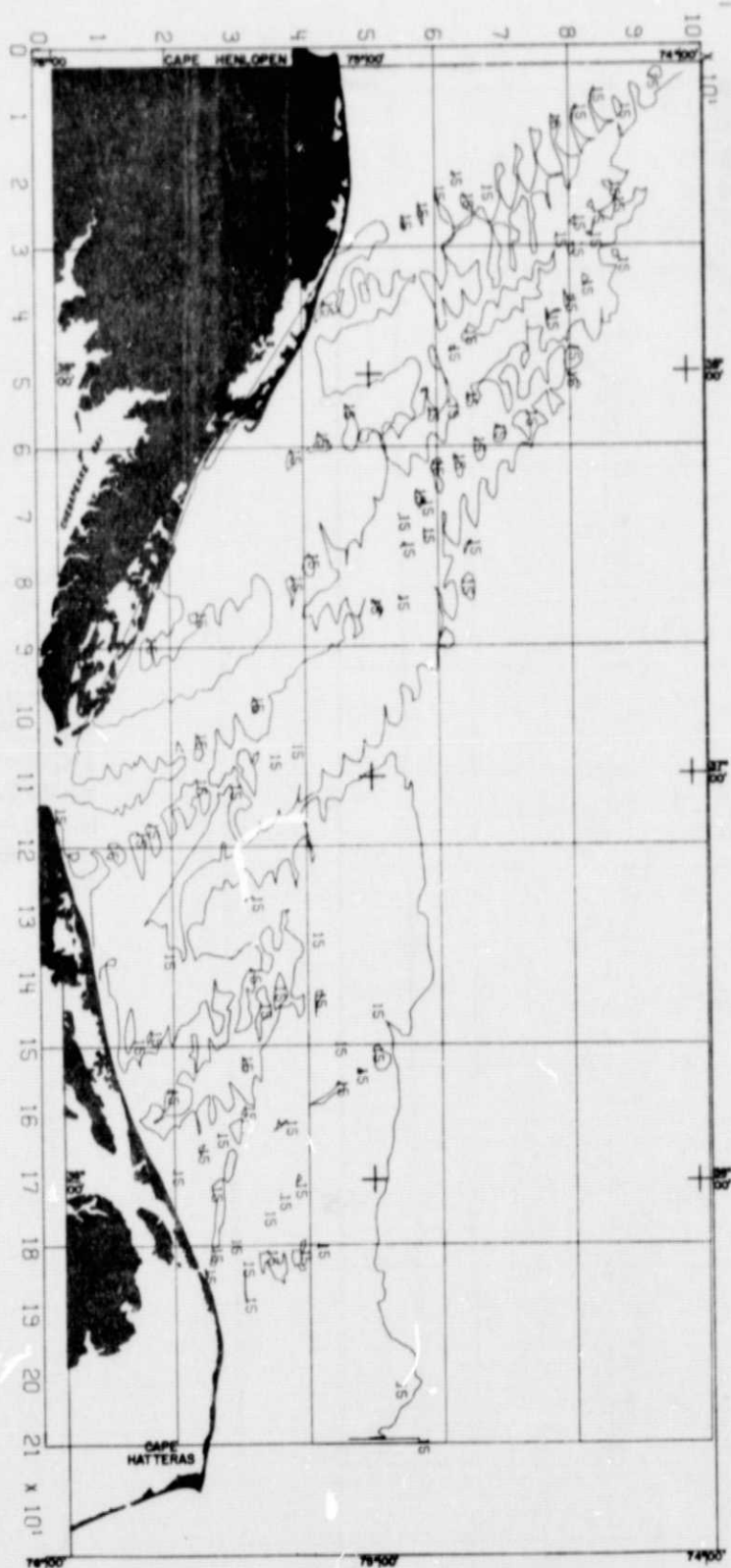


Figure 7. One contour of wave height smoothed with spline under tension.



Namelist input:

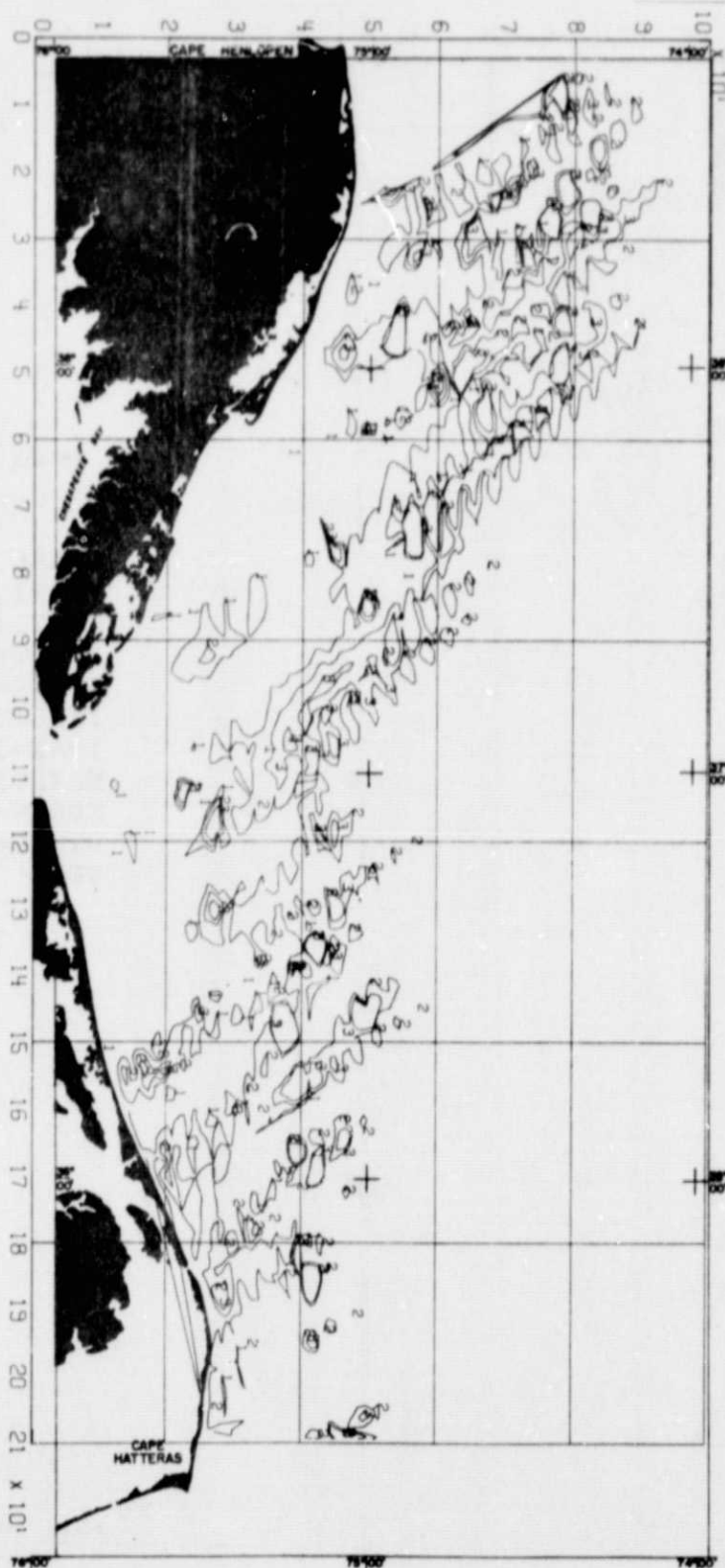
```

$CASE
NCL=1,
CL=0.15E+02,
IPLOT=1,
ISKIP=0,
IZPLOT=2,
IMAX=1000,
MAXPTS=0,
NDEC=-1,
HTNUM=0.15E+00,
$END

```

ORIGINAL PAGE 3  
OF FOUR QUAL 4

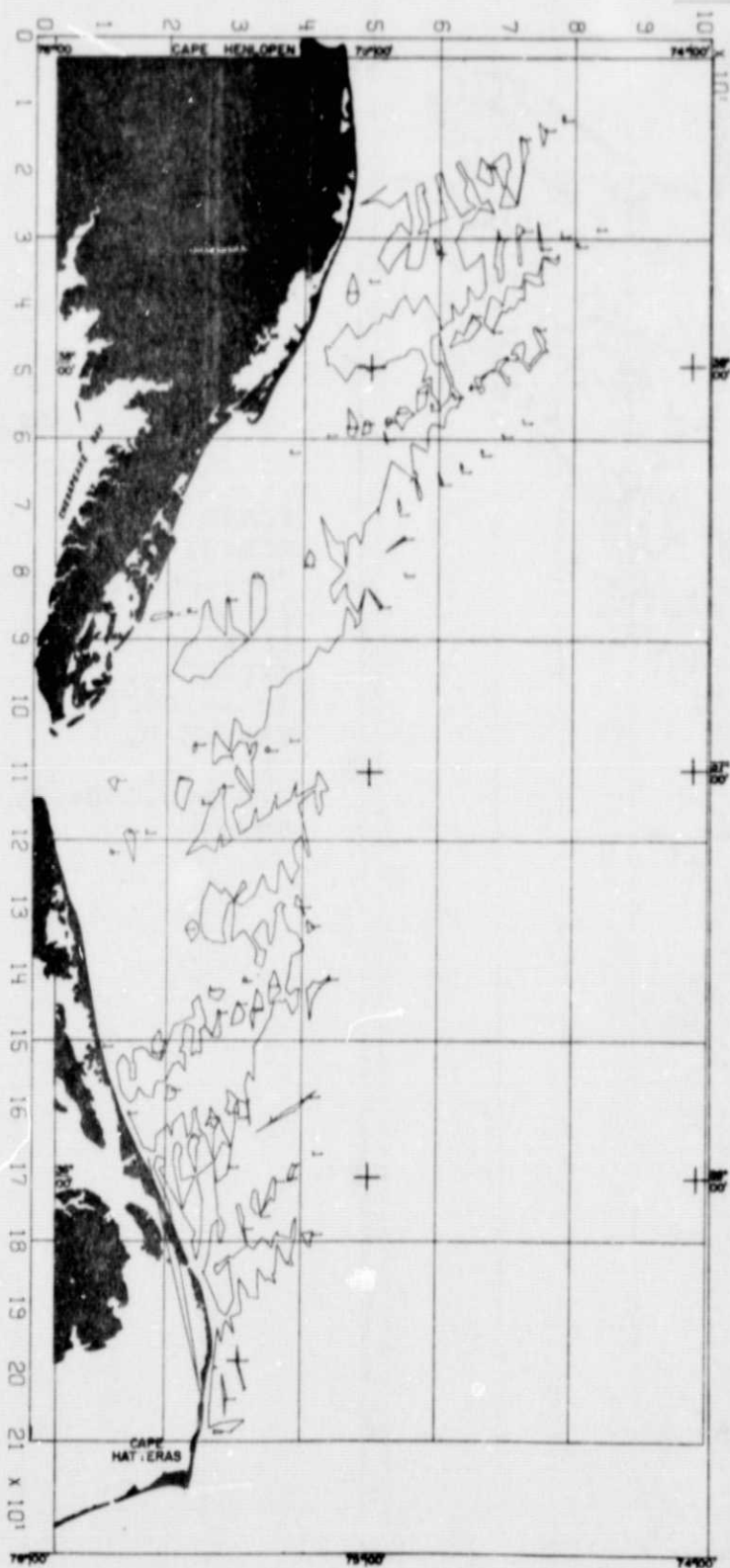
Figure 8. One contour of orbital velocity smoothed with spline under tension.



Namelist input:

```
$CASE
NCL=3,
CL=0.1E+01,0.2E+01,0.3E+01,
IPLOT=1,
ISKIP=0,
IZPLOT=1,
IMAX=1000,
MAXPTS=0,
NDEC=-1,
HTNUM=0.15E+00,
$END
```

Figure 9. Three contours of wave height smoothed with spline under tension.



Namelist input:

```

$CASE
NCL=1,
CL=0.1E+01,
IPLOT=0,
ISKIP=0,
IZPLOT=1,
IMAX=1000,
MAXPTS=0,
NDEC=-1,
HTNUM=0.15E+00,
$END

```

Figure 10. One contour of wave heights with polygonal lines.

## APPENDIX

### Description of Primary - Level Subroutines

The primary-level subroutines are those called by the main program and major modules of the respective basic steps of the program. Brief descriptions of the secondary subroutines and system routines are contained in Tables III and IV.

The following is a description and use of each of the primary subroutines.

ORIGINAL SOURCE  
OF INFORMATION

Name: READIN

**Description:** Subroutine READIN is a driver routine for the contour data input and storage routines. Dynamic storage allocation is used and the program field length is defined accordingly.

**Use:** CALL READIN

**Labeled Common:** /RAF/INDEX(101), LNG(20), BND(2,20),  
NBI(2,20), IFET(18), NBLK  
  
/FLAGS/IPLOT, ISKIP, IZPLOT, IMAX, MAXPTS, LPCT

**Procedure:** An input file is created using dynamic storage for the FET and buffer. The data points are then read into the available core (routine GETXYZ). The minimum and maximum values and the scale factors of the X and Y coordinate arrays are computed (routine MINMAX). The arrays are ordered with a major sort on increasing X and a minor sort on decreasing Y, with duplicate values deleted from the set (routine SHELL). The input file is released and a random access file, RAF, is created. The input data are stored in partitioned form on RAF (routine PUTXYZ). The field length is then reduced to the required level to execute the program.

**External Routines:** GETXYZ, MINMAX, SHELL, PUTXYZ, REQFL, ADDFILE, INITFET, SUBFILE, EVICT, OPENMS, LOCF

Name: GETXYZ

**Description:** Subroutine GETXYZ reads the contour input data from a disk or a magnetic tape. The file read must be generated by the LRC output routine RECOUT.

Use: CALL GETXYZ (X, Y, Z, V)

X - An array of X-coordinates of the contour plot.

Y - An array of Y-coordinates of the contour plot.

Z - An array of wave heights.

V - An array of orbital velocities.

Labeled Common: /FLAGS/IPLOT, ISKIP, IZPLOT, IMAX, MAXPTS, LPCT

Procedure: Entries are made into the X,Y,Z and V arrays until the limit of the points to be read is exceeded or an end of file is detected. The capability of skipping a desired number of records is provided. If an end of file is detected before the limit of points is exceeded, the limit is redefined to the number of points on the file.

External Routines: RECIN

Name: MINMAX

Description: Subroutine MINMAX identifies the minimum and maximum values of two arrays, and computes a scale factor for each array.

Use: CALL MINMAX(X,Y)

X - Array of values

Y - Array of values

Labeled Common: /XTXRXI/NB,NPTS,ISCALE,YSCALE,XMIN,YMIN  
XMAX,YMAX,NL,NT,NI

/FLAGS/IPLOT, ISKIP, IZPLOT, IMAX, MAXPTS, LPCT

**Procedure:** The two arrays, X and Y, are searched for the minimum and maximum values of each. The differences between the respective minimum and maximum values are used to compute the scale factors for the X and Y arrays.

**Error Conditions:** If the difference between a minimum and a maximum value is zero, a message is given and the program is terminated.

**Name:** SHELL

**Description:** Subroutine SHELL uses the shell sorting technique to perform a major and minor sort, disregarding duplicate points.

**Use:** CALL SHELL (X,Y,Z,V,N)

X - Array of the major sort.

Y - Array of the minor sort.

Z - Array sorted with respect to X and Y.

V - Array sorted with respect to X and Y.

N - Number of elements per array sorted.

**Procedure:** A shell sorting technique is used to sort the four arrays with a major sort on increasing X and a minor sort on decreasing Y. If duplicate X-Y points are detected one set of points is deleted and the sort is reinitialized.

Name: PUTXYZ

Description: Subroutine PUTXYZ defines partitions for plotting and stores each partition on a random access file.

Use: CALL PUTXYZ (X,Y,Z,V)

X - Array to be partitioned.

Y - Array to be partitioned.

Z - Array to be partitioned.

V - Array to be partitioned.

Labeled Common: /FLAGS/IPLOT,ISKIP,IZPLOT,IMAX,MAXPTS,LPCT  
/RAF/INDEX(101),LNG(20),BND(2,20),  
NBI(2,20),IFET(18),NBLK

Procedure: The number of partitions are defined such that there will be an overlap of points between partitions. The partition sizes are computed to have an equal number of points per partition with an input overlap. For each partition four records are written for the respective arrays. The record lengths and plot boundaries of the overlaps are stored in core.

Error Condition: If the number of partitions exceed the program limit of twenty, the program is terminated.

External Routines: WRITMS

Name: TRIANG

Description: Subroutine TRIANG is the driver routine for the generation of the triangular grid.

Use: CALL TRIANG (X,Y,N,ITRI,ISUP,IUSED)

X - Array of X-coordinated points for the triangular grid.

Y - Array of Y-coordinate points for the triangular grid.

N - Number of points of the X and Y arrays.

ITRI - Work array containing the triangularization information.

ISUP - Dimension of the ITRI array

IUSED - Actual number of data locations required for the ITRI array.

Labeled Common: /XTXRXI/NB,NPTS,XSCALE,YSCALE,XMIN  
YMIN,XMAX,YMAX,NL,NT,NI

Procedure: The boundary points of the triangular grid are defined (routine CONHUL). An initial grid is generated (routine TMESH2) and then an iteration on the grid is performed for improvement (routine TMESH3).

External Routines: CONHUL,TMESH2,TMESH3

Name: CONHUL

Description: Subroutine CONHUL identifies the boundary points for the triangular grid such that the boundary points form a convex hull.

Use: CALL CONHUL (X,Y,NPTS,IP,IP2,MODE)

X - Array of X-coordinate points.

Y - Array of Y-coordinate points.

NPTS - Number of points in the X-Y arrays.

IP - A pointer array of length NPTS.

IP2 - A dummy array.

MODE - An error flag equal to one if the routine terminates normally.

Labeled Common:           /XTXRXI/I1,NP,XSCALE,YSCALE,XMIN  
                              YMIN,XMAX,YMAX,NL,NT,NI

Procedure:                The IP array is initialized pointing to the sorted X-Y arrays. The minimum point is considered the first point of the boundary. Then the successive points are checked by using the cross product, such that the boundary is defined in a counter-clockwise order. The pointer array is swapped so that the first entries contain the indices of the boundary points.

Error Conditions:        MODE = 2 Less than four points. The algorithm requires at least four points, three boundary and one interior point.

MODE - 5 No interior points.

MODE - 6 All values of the X array equal the maximum value.

Name:                    TMESH2

Description:             Subroutine TMESH2 computes an initial triangular grid for the contour plotting.

Use:                     CALL TMESH2 (X,Y,NPTS,IP,LINE,ITRI,MODE)

X - Array of X-coordinate points for the triangular grid.

Y - Array of Y-coordinate points for the triangular grid.

NPTS - Number of points in the X and Y arrays.

IP - Pointer array for the X and Y arrays.

LINE - Array of descriptors of each line

ITRI - Array of descriptors of each triangle  
of the triangular grid with three  
entries per triangle.

MODE - An error flag equal to one if the  
routine terminates normally.

Labeled Common:           /XTXRXI/I1,NP,XSCALE,YSCALE,XMIN,YMIN,  
                          XMAX,YMAX, NL, NT, NI

Procedure:               A centroid of the interior points is computed  
                          and the nearest point is chosen to initialize  
                          the triangular grid. Lines and triangles are  
                          defined from the boundary and the centroid  
                          points. The remaining interior points are  
                          introduced creating a complete triangular grid.

Error Conditions:       MODE = 6 A point is not contained in any  
                          triangle and is omitted from the set.

MODE = 7 Bad data is encountered and the  
          routine is exited.

MODE = 8 A point not included in the boundary  
          set but now lies on the boundary.  
          The point is omitted from the set.

External Routines       . STEST,LTEST

Name :                   TMESH3

Description:            Subroutine TMESH3 performs an iterative  
                          improvement of the triangulation.

Use:                    CALL TMESH3 (X,Y,LINE,ITRI)

X - Array of X-coordinate points for the  
     triangular grid.

Y - Array of Y-coordinate points for the triangular grid.

LINE - Array of descriptors of each triangle of the triangular grid with three entries per triangle.

Labeled Common:           /XTXRXI/NB,NPTS,XSCALE,YSCALE,XMIN,YMIN  
                          XMAX,YMAX,NL,NT,NI

/TMCOM2/LL(4)

Procedure:               An iteration is performed to improve the triangulation by interchanging lines of the triangle so that the smallest angles of the triangles are maximized. A limit of three times the number of lines is imposed on the iteration count.

External Routine:       STEST

Name:                    TRIORD

Description:            Subroutine TRIORD reorders the line indices with respect to the Z vector, such that moving from the initial to the terminal point on a line involves an increase in the associated Z values.

Use:                    CALL TRIORD (Z,N,ITRI,ZSPEC)

Z - Array associated with the line and triangle indices.

N - Number of values in the Z array.

ITRI - Array containing the line and triangle information.

ZSPEC - A value of the Z array which is to be ignored.

Labeled Common:        /XTXRXI/NB,NPTS,XSCALE,YSCALE,XMIN,YMIN,  
                          XMAX,YMAX,NL,NT,NI

Procedure: For each of the lines, the associated Z values of the end points are reordered such that the terminal Z value is greater than the initial Z value. Sign changes are made on the corresponding triangle data if there is a reordering.

Name: SCAN

Description: Subroutine SCAN produces the ordered sequence of X-Y pairs of points on a given contour.

Use: CALL SCAN (X,Y,Z,N,ITRI,CL)

X - Array of X coordinates of the surface data.

Y - Array of Y coordinates of the surface data.

Z - Array of Z coordinates of the surface data.

N - Number of data points.

ITRI - Array containing the line and triangle information.

CL - Value of the contour.

Labeled Common: /XTXRXI/NB,NPTS,XSCALE,YSCALE,XMIN  
YMIN,XMAX,YMAX,NL,NT,NI

Procedure: For each line, a check is made for an intersection. The contour is continued by checking the lines of adjacent triangles until the contour is enclosed or intersects a boundary line. If the boundary is intersected, the contour is continued backwards from the first point until the second boundary is encountered. For each line crossed, a linear interpolation between the end points is made and commands are given to plot the computed point.

External Routines: FRSTPT,VECTOR

Name: CONTRI

Description: Subroutine CONTRI checks the triangular grid data and initializes the contour arrays for the spline with tension curve fitting.

Use: CALL CONTRI (X,Y,Z,LINE,ITRI,CTRVAL,NC,MODE)

X - Array of X-coordinate points.

Y - Array of Y-coordinate points.

Z - Array of Z-coordinate points.

LINE - Array of descriptors of each line of the triangular grid with five entries per line.

ITRI - Array of descriptors of each triangle of the triangular grid with three entries per line.

CTRVAL - Array of contour levels to be plotted.

NC - Number of contour levels.

MODE - An error flag equal to one if the routine terminates normally.

Labeled Common: /XTYRXI/I1,NP,XSCALE,YSCALE,XMIN, YMIN,  
XMAX,YMAX,NL,NT,NI

/CONTROL/CLEVEL,NDEC,HTN

Internal Data: XPS - Array of interpolated X values associated with a contour level, dimensioned 500.

YPS - Array of interpolated Y values associated with a contour level, dimensioned 500.

Procedure: The line descriptor array is reordered such that the end points of the line are increasing with respect to the Z values. For each contour level, the line descriptor array is searched for an intersection, defining a starting point of a contour. A neighboring triangle is used to continue the contour until a boundary is reached

or the contour is closed. When a line is used for a contour, it is flagged so it will not be used a second time. Arrays are generated from the interpolated X-Y values of the end points of the lines crossed to be used in the spline with tension curve fitting routines (routine CURVL).

Error Conditions:      MODE = 7 Bad data structure. A line does not have an associated triangle.

                         MODE = 8 Bad data structure. A contour level crossing one side of a triangle does not cross another.

                         MODE = 9 Too many lines crossed by a contour. The XPS and YPS arrays are insufficiently dimensioned.

External Routine:      CURVL

Name:                      CURVL

Description:              Subroutine CURVL determines if the curve is closed or open and calls the appropriate spline routines.

Use:                        CALL CURVL (X,Y,NPPTS,NNPTS,FAC)

                         X - Array of X-coordinate points to be curve fitted.

                         Y - Array of Y-coordinate points to be curve fitted.

                         NPPTS - Number of points of the contour before a boundary line is encountered in the arrays.

                         NNPTS - Number of points of the contour after a boundary line is encountered in the arrays.

                         FAC - The magnitude of the plot.

Internal Data:            XP - Array for the storage of curvature information for the given nodes, dimensioned 500.

SCRAT - Temporary storage array, dimensioned 1000.

Labeled Common:               /SIG1/SIGMA

Procedure                   If the arrays define a closed contour,  
the closed curve routines are executed  
(KURVP1 and KURVP2). If the contour is  
open, the arrays are reordered such that  
the first and last points are the end  
points, and the open and curve routines  
are executed (KURV1 and KURV2). When  
a point is defined the plotting routines  
are executed.

External Routines:       KURV1, KURV2, KURVP1, KURVP2, POINT, PRSTPT,  
VECTOR.